1503.64973 PATENT APPLICATION

RAB 8-26-02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re U.S. Patent Application)
)
Applicant: Akira Tsuboi)
)
Serial No.)
)
Filed: Januaray 21, 2000)
)
For: APPARATUS AND METHOD)
FOR EXECUTING PROGRAM)
USING JUST-IN-TIME)
COMPILER SYSTEM)
Art Unit:)

I hereby certify that this paper is being deposited with the United States Postal Service as EXPRESS mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231, on December 8, 2000.
Express Label No.: EL 769180941 US
Signature: _____

CLAIM FOR PRIORITY

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Applicant claims foreign priority benefits under 35 U.S.C. § 119 on the basis of the foreign application identified below:

Japanese Patent Application No. 2000-012599, filed January 21, 2000.

A certified copy of the priority document is enclosed.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By: _____
James K. Folker
Reg. No. 37,538

December 8, 2000
300 South Wacker Drive
Suite 2500
Chicago, IL 60606
(312) 360-0080

1503.64973
(312)360-0083

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 １月２１日

出 願 番 号
Application Number: 特願２０００－０１２５９９

出 願 人
Applicant (s): 富士通株式会社

２０００年 ９月 ８日

特 許 庁 長 官
Commissioner,
Patent Office 及 川 耕 造

出証番号 出証特２０００－３０７１４２４

【書類名】　特許願
【整理番号】　9951783
【提出日】　平成12年 1月21日
【あて先】　特許庁長官殿
【国際特許分類】　G06F　9/45
【発明の名称】　プログラム実行装置、プログラム実行方法、及び記録媒体
【請求項の数】　10
【発明者】
【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
【氏名】　坪井　晃
【特許出願人】
【識別番号】　000005223
【氏名又は名称】　富士通株式会社
【代理人】
【識別番号】　100074099
【住所又は居所】　東京都千代田区二番町８番地２０　二番町ビル３Ｆ
【弁理士】
【氏名又は名称】　大菅　義之
【電話番号】　03-3238-0031
【選任した代理人】
【識別番号】　100067987
【住所又は居所】　神奈川県横浜市鶴見区北寺尾７－２５－２８－５０３
【弁理士】
【氏名又は名称】　久木元　彰
【電話番号】　045-573-3683
【手数料の表示】
【予納台帳番号】　012542

【納付金額】　　21,000円

【提出物件の目録】

【物件名】　　明細書　1

【物件名】　　図面　1

【物件名】　　要約書　1

【包括委任状番号】　9705047

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　プログラム実行装置、プログラム実行方法、及び記録媒体

【特許請求の範囲】

【請求項１】　機械語を実行する実行手段を有し、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを該実行手段で直接実行可能な機械語に翻訳して実行するプログラム実行装置であって、

前記原始プログラムに記述される関数の翻訳であって前記実行手段で実行可能な機械語を該関数ごとに記憶する、電源電圧が消失しても記憶内容が保持される記憶手段と、

前記原始プログラムを前記実行手段で実行可能な機械語に翻訳する翻訳手段と、

前記翻訳手段により翻訳された機械語を前記記憶手段に記憶させる記憶制御手段と、

前記原始プログラムで用いられている関数の翻訳である機械語が前記記憶手段に記憶されているか否かの判定を行なう判定手段と、

前記判定手段による判定結果に応じて、前記翻訳手段の翻訳する機械語、または前記記憶手段に記憶されている機械語、のどちらかを前記実行手段に直接実行させる実行制御手段と、

を有することを特徴とするプログラム実行装置。

【請求項２】　前記記憶手段には、前記原始プログラムで用いられることのあり得る関数の翻訳である機械語が予め記憶されていることを特徴とする請求項１に記載のプログラム実行装置。

【請求項３】　前記記憶手段の記憶内容を複写して記憶する半導体メモリを更に有し、

前記実行制御手段は、前記記憶手段に記憶されている機械語を前記実行手段に実行させる代わりに、前記半導体メモリに記憶されている、該記憶手段に記憶されている記憶内容の複写である機械語を該実行手段に実行させる、

ことを特徴とする請求項１に記載のプログラム実行装置。

【請求項４】　機械語を実行する実行手段を有し、ジャスト・イン・タイム

・コンパイラ方式により、原始プログラムを該実行手段で直接実行可能な機械語に翻訳して実行するプログラム実行装置であって、

前記原始プログラムに記述される関数の翻訳であって前記実行手段で実行可能な機械語を該関数ごとに記憶し、該原始プログラムの実行終了後も記憶内容を保持する記憶手段と、

前記原始プログラムを前記実行手段で実行可能な機械語に翻訳する翻訳手段と、

前記翻訳手段により翻訳された機械語を、前記翻訳手段により翻訳される原始プログラムが更新された日時に対応付けて、前記記憶手段に記憶させる記憶制御手段と、

前記原始プログラムの更新された日時と、前記記憶手段に記憶されている前記機械語に対応付けられている更新日時とが一致するか否かを判定する判定手段と、

前記判定手段による判定結果に応じて、前記翻訳手段の翻訳する機械語、または前記記憶手段に記憶されている機械語、のどちらかを前記実行手段に直接実行させる実行制御手段と、

を有することを特徴とするプログラム実行装置。

【請求項５】　前記原始プログラムが格納されているプログラムファイルを読み込む読込手段を更に有し、

前記記憶制御手段は、前記プログラムファイルに示されている該プログラムファイルの更新日時を前記機械語に対応付ける該原始プログラムの更新された日時とみなして、前記記憶手段に該機械語を記憶させ、

前記判定手段は、前記プログラムファイルに示されている該プログラムファイルの更新日時と、前記機械語に対応させて前記記憶手段に記憶されている更新日時とが一致するか否かを判定する、

ことを特徴とする請求項４に記載のプログラム実行装置。

【請求項６】　前記原始プログラムは、Ｊａｖａのバイト・コードで記述されていることを特徴とする請求項１又は４に記載のプログラム実行装置。

【請求項７】　ジャスト・イン・タイム・コンパイラ方式により、原始プロ

グラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させるプログラム実行方法であって、

前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、電源電圧が消失しても記憶内容が保持される記憶部に記憶させ、

前記原始プログラムに記述されている関数の翻訳である前記機械語が前記記憶部に記憶されているか否かの判定を行ない、

前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または前記記憶部に記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させる、

ことを特徴とするプログラム実行方法。

【請求項８】　ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させるプログラム実行方法であって、

前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、該機械語の翻訳前の原始プログラムが更新された日時に対応付けて記憶し、

前記原始プログラムの更新された日時と、記憶されている前記機械語に対応付けられている更新日時とが一致するか否かの判定を行ない、

前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させる、

ことを特徴とするプログラム実行方法。

【請求項９】　コンピュータに実行させることによって、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させる制御を該コンピュータに行なわせる制御プログラムを記録したコンピュータで読み取り可能な記録媒体であって、

前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述され

ている関数ごとに、電源電圧が消失しても記憶内容が保持される記憶部に記憶させるステップと、

前記原始プログラムに記述されている関数の翻訳である前記機械語が前記記憶部に記憶されているか否かの判定を行なうステップと、

前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または前記記憶部に記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させるステップと、

からなる処理を該コンピュータに行なわせる制御プログラムを記録した記録媒体。

【請求項１０】　コンピュータに実行させることによって、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させる制御を該コンピュータに行なわせる制御プログラムを記録したコンピュータで読み取り可能な記録媒体であって、

前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、該機械語の翻訳前の原始プログラムが更新された日時に対応付けて記憶するステップと、

前記原始プログラムの更新された日時と、記憶されている前記機械語に対応付けられている更新日時とが一致するか否かの判定を行なうステップと、

前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させるステップと、

からなる処理を該コンピュータに行なわせる制御プログラムを記録した記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、高水準のプログラミング言語で記述されているプログラムを演算処理システムで実行させる技術に関し、特に、ジャスト・イン・タイム・コンパイ

ラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させる技術に関する。

【０００２】

【従来の技術】

異なる演算処理システムのプラットホーム上で同一のプログラムを動作させようとする多くの試みが以前から行なわれている。プログラミング言語Ｊａｖａ（Ｊａｖａは、サン・マイクロシステムズ・インコーポレイテッドの登録商標）はその回答のひとつである。Ｊａｖａはプラットホーム独立を実現し、異種のプラットホーム間での可搬性に優れている。

【０００３】

Ｊａｖａで記述されたプログラムのソースコードは、構文解析などの処理が施され、バイトコードと呼ばれるバイナリファイルに変換されて配布されるのが一般的である。Ｊａｖａバイトコードは演算処理システムのプラットホームに独立である汎用的な命令コードであり、Ｊａｖａ　ＶＭ（バーチャル・マシン）と呼ばれるＪａｖａの実行系で解釈され実行されるコードである。Ｊａｖａ　ＶＭに相当する環境を各々の演算処理システムが備えることによって、異なる演算処理システム上で同一のＪａｖａバイトコードの実行が可能となる。

【０００４】

以下、Ｊａｖａバイトコードを演算処理システム上で実行させる技術について説明する。

【０００５】

Ｊａｖａインタプリタは、Ｊａｖａバイトコードを命令毎に逐次解釈（インタプリト）し、演算処理システムにその命令に対応する処理を行なわせるものである。Ｊａｖａインタプリタは、Ｊａｖａバイトコードで記述されたプログラムをそのまま解釈して処理できることが大きな利点ではあるが、命令解釈のための時間を要するために処理速度が遅いという短所がある。

【０００６】

Ｊａｖａバイトコードで記述されたプログラムの実行速度を向上させる手法として、処理させる演算処理システムで直接実行可能な機械語であるネイティブコ

ードにＪａｖａバイトコードを予め翻訳（コンパイル）しておく手法があり、このためのツールをネイティブ・コンパイラという。Ｊａｖａバイトコードのネイティブコード化によりプログラムの実行速度は格段に向上するが、Ｊａｖａの大きな利点である異種のプラットホーム間での可搬性が損なわれるという問題があり、更に、プログラムの変更によりＪａｖａバイトコードが更新される度に、演算処理システムのオペレータがネイティブコードへの翻訳作業をシステムに行なわせるように指示しなければならない煩わしさもある。

【０００７】

Ｊａｖａの有する利点である異種のプラットホーム間での可搬性を維持しつつ、Ｊａｖａインタプリタの抱える処理速度の問題を改善するためにジャスト・イン・タイム・コンパイラ（以下、「ＪＩＴコンパイラ」という）が提案されている。

【０００８】

ＪＩＴコンパイラは、Ｊａｖａバイトコードを実行する際に、そのバイトコードで用いられている関数　（Ｊａｖａにおけるメソッド）を単位としてネイティブコードに翻訳しながら演算処理システムに逐次実行させる。ここで、翻訳されたネイティブコードをメインメモリに保持しておくようにする。Ｊａｖａバイトコードの翻訳を進めていくうちに、既にネイティブコードに翻訳されている関数が出現したときは、その関数についての再度の翻訳は行なわずに、メインメモリに保持されているネイティブコードをそのまま演算処理システムに直接実行させるようにして翻訳処理に要する処理時間を削減する。一般的にプログラム全体の実行速度を低下させる大きな影響を与えている関数は繰り返し呼び出されていることが多々あるので、この手法を用いることによりＪａｖａバイトコードの実行速度を向上させることができる。

【０００９】

また、ＪＩＴコンパイラは、インタプリタによる解釈処理では困難なＪａｖａバイトコードで記述されている命令の最適化も行なうので、この最適化によっても処理速度が高速化される。しかも、ネイティブ・コンパイラでは必要となる、演算処理システムに対するオペレータによる煩雑な操作はＪＩＴコンパイラでは

不要である。

【００１０】

なお、ＪＩＴコンパイラは、Ｊａｖａバイトコードの実行に使用するものが特に広く普及しているが、他のプログラミング言語で記述されたプログラムや、プログラムに対して構文解析や最適化の処理が施されて得られる中間言語で記述されているプログラム（本明細書では、これらのプログラムを総称して「原始プログラム」と称することとする）を、特定のプラットホームが直接解釈できるネイティブコードに翻訳しながらそのプラットホーム上で実行させるように構成することも可能である。

【００１１】

【発明が解決しようとする課題】

前述したように、ＪＩＴコンパイラは、Ｊａｖａの有する大きな利点である異種のプラットホーム間での可搬性を維持しつつ実行速度の問題を改善することが可能であり、更にオペレータによる操作作業の負担がネイティブ・コンパイラよりも少ない利点を有している。しかしながら、原始プログラムの実行開始時にネイティブコードへの翻訳処理が必ず行なわれるため、ＪＩＴコンパイラを用いて行なう原始プログラムの実行では、その原始プログラムに記述されている処理が実際に開始されるまでにある程度のタイムラグが生じてしまう問題を有していた。

【００１２】

以上の問題を鑑み、ＪＩＴコンパイラを用いて原始プログラムを実行させるときの実行開始時の性能を向上させることが本発明が解決しようとする課題である。

【００１３】

【課題を解決するための手段】

図１は本発明の基本構成図である。本発明に係る装置であるプログラム実行装置１０は、機械語を実行する実行手段１６を有し、ジャスト・イン・タイム・コンパイラ方式により、原始プログラム１を実行手段１６で直接実行可能な機械語に翻訳して実行する装置を前提とする。

【００１４】

本発明の第一の態様の装置では、プログラム実行装置１０に、原始プログラム１に記述される関数の翻訳であって実行手段１６で実行可能な機械語を該関数ごとに記憶する、電源電圧が消失しても記憶内容が保持される記憶手段１１と、原始プログラム１を実行手段１６で実行可能な機械語に翻訳する翻訳手段１２と、翻訳手段１２により翻訳された機械語を記憶手段１１に記憶させる記憶制御手段１３と、原始プログラム１で用いられている関数の翻訳である機械語が記憶手段１１に記憶されているか否かの判定を行なう判定手段１４と、判定手段１４による判定結果に応じて、翻訳手段１２の翻訳する機械語、または記憶手段１１に記憶されている機械語、のどちらかを実行手段１６に直接実行させる実行制御手段１５とを有するように構成することによって前述した課題を解決する。

【００１５】

この構成においては、記憶手段１１としては、例えばハードディスク装置やフラッシュＥＥＰＲＯＭ（一括消去型電気的消去及び書き込み可能読み出し専用半導体メモリ）などを用いることができる。

【００１６】

上記の構成では、記憶手段１１が原始プログラム１に記述される関数の翻訳である機械語を記憶する。原始プログラム１で用いられている関数の翻訳である機械語が記憶手段１１に記憶されていると判定手段１４が判定すれば、実行制御手段１５は翻訳手段１２によるその関数の翻訳を待たずに、記憶手段１１に記憶されているその関数の機械語を直接実行させるように実行手段１６を制御するので、プログラム実行装置１０は前述したＪＩＴコンパイラに相当する動作が行なわれる。従って、上記の構成によれば、ＪＩＴコンパイラの有する前述した利点をそのまま有している。

【００１７】

しかも、上記の構成における記憶手段１１は、電源電圧が消失してもその記憶内容が保持される。従って、例えばオペレータが一日の仕事を終えてプログラム実行装置１０の電源を切り、その翌日に仕事を再開するような場合、前日までに同一の原始プログラム１を一度でも実行していれば、記憶手段１１に記憶されて

いる、原始プログラム１に記述されている関数の翻訳である機械語をその日の最初の実行から利用することができるので、プログラム実行装置１０はプログラムの翻訳処理に起因する実行開始時のタイムラグを生じさせずに原始プログラム１を実行させることができる。

【００１８】

なお、上述した構成において、記憶手段１１には、原始プログラム１で用いられることのあり得る関数の翻訳である機械語が予め記憶されているように構成しても良い。この構成では、プログラム実行装置１０上での原始プログラム１の初めての実行の際に原始プログラム１に記述されている関数の翻訳である機械語が既に記憶されている場合があり、このような場合には原始プログラム１に記述されている関数の翻訳である機械語を記憶手段１１から得ることができるので、プログラムの翻訳処理に起因する実行開始時のタイムラグを短縮することができる。

【００１９】

また、上述した構成において、記憶手段１１の記憶内容を複写して記憶する半導体メモリを更に有し、実行制御手段１５は、記憶手段１１に記憶されている機械語を実行手段１６に実行させる代わりに、その半導体メモリに記憶されている、記憶手段１１に記憶されている記憶内容の複写である機械語を実行手段１６に実行させるように構成しても良い。例えば記憶手段１１としてハードディスク装置を用いている場合などでは、ハードディスク装置から機械語を読み出すよりも、そのハードディスク装置と同一の記憶内容が複写されている半導体メモリから機械語を読み出す方が、機械語の検索・読み出しが高速に行なえるので、プログラム装置１０による原始プログラム１の翻訳・実行の処理時間を更に短縮することができる。

【００２０】

また、本発明の第二の態様の装置では、プログラム実行装置１０に、原始プログラム１に記述される関数の翻訳であって実行手段１６で実行可能な機械語を該関数ごとに記憶し、原始プログラム１の実行終了後も記憶内容を保持する記憶手段１１と、原始プログラム１を実行手段１６で実行可能な機械語に翻訳する翻訳

手段１２と、翻訳手段１２により翻訳された機械語を、翻訳手段１２により翻訳される原始プログラム１が更新された日時に対応付けて、記憶手段１１に記憶させる記憶制御手段１３と、原始プログラム１の更新された日時と、記憶手段１１に記憶されている前記機械語に対応付けられている更新日時とが一致するか否かを判定する判定手段１４と、判定手段１４による判定結果に応じて、翻訳手段１２の翻訳する機械語、または記憶手段１１に記憶されている機械語、のどちらかを実行手段１６に直接実行させる実行制御手段１５とを有するように構成することによって前述した課題を解決する。

【００２１】

上記の構成は、原始プログラム１が格納されているプログラムファイルを読み込む読込手段を更に有し、記憶制御手段１３は、原始プログラム１が格納されているプログラムファイルに示されている該プログラムファイルの更新日時を前記機械語に対応付ける原始プログラム１の更新された日時とみなして、記憶手段１１に該機械語を記憶させ、判定手段１４は、そのプログラムファイルに示されている該プログラムファイルの更新日時と、前記機械語に対応させて記憶手段１１に記憶されている更新日時とが一致するか否かを判定するように構成してもよい。

【００２２】

上記の構成では、前述した本発明の第一の態様の装置と同様に、ＪＩＴコンパイラの有する利点をそのまま有している。

【００２３】

更に、上記の構成では、原始プログラム１の実行終了後も記憶内容を保持しているので、同一の原始プログラム１を再度実行させる場合には、実行制御手段１５は翻訳手段１２による原始プログラム１の翻訳を待たずに、記憶手段１１に記憶されているその関数の機械語を実行手段１６に直接実行させることにより、プログラムの翻訳処理に起因する実行開始時のタイムラグを生じさせずに原始プログラム１を実行させることができる。ただし、この場合、後に実行される原始プログラム１と先に実行されたものとが、原始プログラムのバージョンアップによる修正等により、同一でないことがあり得る。そこで、記憶制御手段１３が、翻

訳手段１２により翻訳された機械語と、翻訳手段１２により翻訳される原始プログラム１が更新された日時とを対応付けて記憶手段１１に記憶させ、判定手段１４が、原始プログラム１の更新された日時と、機械語に対応付けて記憶手段１１に記憶されている更新日時とが一致するか否かを判定する。そしてこの判定の結果、両者が一致しないのであれば、原始プログラム１で用いられている関数の翻訳である機械語が記憶手段１１に記憶されていても、実行制御手段１５は翻訳手段１２により新たに翻訳された機械語を実行手段１６に実行させる。こうすることによって、原始プログラム１に変更が加えられても、その変更に応じて正しく原始プログラム１を実行させることができる。

【００２４】

なお、上記の構成においては、記憶手段１１として、ハードディスク装置やフラッシュＥＥＰＲＯＭなどの、電源電圧が消失しても記憶内容が保持されるものに限らず、他の読み書き可能な記憶媒体を使用することもできる。

【００２５】

なお、上述した本発明の第一または第二の態様の装置において、原始プログラム１は、Ｊａｖａのバイト・コードで記述されていてもよい。この場合、原始プログラム１に記述される関数とはＪａｖａにおけるメソッドに相当する。

【００２６】

また、本発明に係るプログラム実行方法は、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させる方法を前提とする。

【００２７】

そして、本発明の第一の態様のプログラム実行方法では、前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、電源電圧が消失しても記憶内容が保持される記憶部に記憶させ、前記原始プログラムに記述されている関数の翻訳である前記機械語が前記記憶部に記憶されているか否かの判定を行ない、前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または前記記憶部に記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させることによって前述した課題を

解決する。そして、このプログラム実行方法によっても前述した本発明の第一の態様の装置と同様な作用・効果を奏する。

【００２８】

また、本発明の第二の態様のプログラム実行方法では、前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、該機械語の翻訳前の原始プログラムが更新された日時に対応付けて記憶し、前記原始プログラムの更新された日時と、記憶されている前記機械語に対応付けられている更新日時とが一致するか否かの判定を行ない、前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させることによって前述した課題を解決する。そして、このプログラム実行方法によっても前述した本発明の第二の態様の装置と同様な作用・効果を奏する。

【００２９】

なお、上述した本発明の各構成により行なわれる機能と同様の制御をコンピュータに行なわせる制御プログラムを記録したコンピュータ読み取り可能な記録媒体から、その制御プログラムをコンピュータに読み取らせて実行させることによっても、前述した課題を解決することができ、この構成によっても前述したプログラム実行装置と同様な作用・効果を奏する。

【００３０】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づいて説明する。なお、ここでは、Ｊａｖａのバイトコードを翻訳して実行するプログラム実行装置において本発明を実施する例について説明する。

【００３１】

図２は本発明を実施するプログラム実行装置の全体構成を示す図である。同図に示すように、このプログラム実行装置（以下、「本装置」という）２０は、入力部２１、ＣＰＵ２２、出力部２３、Ｉ／Ｆ部２４、半導体メモリ２５、ハードディスク装置２６を有し、バス２７を介して相互に接続されている。

【００３２】

入力部２１は、キーボード装置や、マウス等のポインティングデバイスなどを有し、本装置のオペレータからの各種指示の入力を取得する。また、フロッピーディスクや光磁気ディスク、磁気テープなどの記録媒体からデータを読み出すためのデータ読み出し装置を有するようにしてもよい。

【００３３】

ＣＰＵ２２は半導体メモリ２５に記憶された制御プログラムに従って本装置２０全体の動作を制御する中央演算処理装置であり、機械語を直接実行する。

【００３４】

出力部２３は、ディスプレイ装置やプリンタなどを有し、本装置２０で行なった処理の結果等をオペレータに提示するものである。

【００３５】

Ｉ／Ｆ部２４は、本装置２０を他の装置やネットワークに接続するためのインターフェースのための処理を司るものである。

【００３６】

半導体メモリ２５は、本装置２０全体の制御処理をＣＰＵ２２に行なわせるための制御プログラムを予め記憶しておいたり、ハードディスク装置２６の記憶内容を複写して記憶したり、あるいはＣＰＵ２２により実行される処理のためのワークエリアとして用いられるものであり、ＲＯＭ（リード・オンリ・メモリ）２５－１及びＲＡＭ（ランダム・アクセス・メモリ）２５－２を有している。

【００３７】

ハードディスク装置２６は、本装置２０へ供給される電源電圧が消失しても記憶内容が保持されるデータ記憶装置である。

【００３８】

次に図３について説明する。図３は、本装置２０における翻訳・実行処理の全体の流れを図で示したものである。

【００３９】

プログラムソースモジュール４１は、本装置２０に翻訳・実行を行なわせるソフトウェアモジュールであり、本実施の形態ではＪａｖａのバイトコードによって記述されている。プログラムソースモジュール４１は、前述した記録媒体に格

納されているプログラムファイルとして提供されたものを入力部２１のデータ読み取り装置で読み取ることにより、あるいは、他の装置やネットワークからプログラムファイルとして送付されたものをＩ／Ｆ部２４が受け取ることにより、本装置２０に取り込まれる。

【００４０】

本装置２０に入力されたプログラムソースモジュール４１は、ＪＩＴコンパイラ３０により翻訳・実行される。ＪＩＴコンパイラ３０は、ＣＰＵ２２が本装置２０全体の制御プログラムを実行することにより実現される。

【００４１】

ＪＩＴコンパイラ３０によって行なわれる処理の内容を機能別に分けると、プログラムロード部３１、高速化判定部３２、コンパイル（翻訳）部３３、ファイルロード部３４、プログラム実行部３５の各機能ブロックに大別される。

【００４２】

プログラムロード部３１は、本装置２０に取り込まれたプログラムファイルからプログラムソースモジュール４１を取り出して、ＲＡＭ２５－２におけるＣＰＵ２２がワークメモリとして使用している領域（ワークエリア）に格納する。

【００４３】

高速化判定部３２は、プログラムソースモジュール４１の構文分析を行ない、そこに記述されているメソッドの定義の翻訳に相当するネイティブコード（コンパイル済み格納モジュール４３）が、ハードディスク装置２６に記憶されているか否かを、後述する管理簿４２を参照することにより判定する。本実施の形態では、管理簿４２はハードディスク装置２６に記憶されている。

【００４４】

コンパイル部３３は、上述したメソッドの定義の翻訳に相当するコンパイル済み格納モジュール４３がハードディスク装置２６に記憶されていないと高速化判定部３２が判定したときに、そのメソッドの定義の翻訳を行ない、その結果得られたＣＰＵ２２で直接実行可能なネイティブコードを、ＲＡＭ２５－２における前述したワークエリアにコンパイル済み実行モジュール４４として格納すると共に、コンパイル済み格納モジュール４３としてハードディスク装置２６に記憶さ

せる。更に、管理簿４２におけるその記憶させたコンパイル済み格納モジュール４３についての管理情報を更新する。

【００４５】

ファイルロード部３４は、上述したメソッドの定義の翻訳に相当するコンパイル済み格納モジュール４３がハードディスク装置２６に記憶されていると高速化判定部３２が判定したときに、そのコンパイル済み格納モジュール４３をハードディスク装置２６からロードし（読み出し）、ＲＡＭ２５－２のワークエリアにコンパイル済み実行モジュール４４として格納する。

【００４６】

プログラム実行部３５は、コンパイル部３３もしくはファイルロード部３４によってＲＡＭ２５－２のワークエリアに格納されたコンパイル済み実行モジュール４４をＣＰＵ２２で直接実行する。

【００４７】

ＪＩＴコンパイラ３０は、これらの高速化判定部３２、コンパイル（翻訳）部３３、ファイルロード部３４、プログラム実行部３５の各機能ブロックで示される処理を繰り返し実行することによって、プログラムソースモジュール４１を実行させる。

【００４８】

次に管理簿４２について説明する。図４は管理簿４２の一例を表として示す図である。管理簿４２は、ハードディスク装置２６に記憶されているコンパイル済み格納モジュール４３についての管理情報が記録されるものである。

【００４９】

図４に示す表において、左側の欄には、ハードディスク装置２６に記憶させたコンパイル済み格納モジュール４３に対応する、そのコンパイル済み格納モジュール４３が翻訳である翻訳前のメソッド定義のメソッド名が記録される。そして、同図に示す表の右側の欄には、左側の欄のメソッド名に対応するメソッド定義が記述されていたプログラムソースモジュール４１の更新日時が記録される。この更新日時には、本実施の形態では、プログラムソースモジュール４１が格納されていたプログラムファイルの管理情報として示されている更新日時を用いる。

高速化判定部３２では、プログラムロード部３１がＲＡＭ２５－２のワークエリアに格納したプログラムソースモジュール４１を分析する。そして、そこに記述されているメソッド定義のメソッド名が管理簿４２に記録されており、且つそのプログラムソースモジュール４１の更新日時が管理簿４２のそのメソッド名に対応する更新日時と一致するか否かを判定する。この判定結果が真であれば、そのメソッドの定義の翻訳に相当するコンパイル済み格納モジュール４３がハードディスク装置２６に記憶されていると判定する。

【００５０】

次に図５について説明する。図５は、ＲＯＭ２５－１に格納されている、本装置２０全体の制御プログラムをＣＰＵ２２が実行することにより行なわれる制御処理のうち、本発明に関係する、プログラムソースモジュール４１の翻訳・実行を行なう処理であるＪＩＴコンパイル処理の第一の例の処理内容をフローチャートで示した図である。以下、同図に沿ってＣＰＵ２２が行なうＪＩＴコンパイル処理について説明する。

【００５１】

まず、ＣＰＵ２２は本装置２０に取り込まれたプログラムファイルからプログラムソースモジュール４１を取り出し、ＲＡＭ２５－２のワークエリアに格納する（Ｓ１０１）。この処理は図３におけるプログラムロード部３１の機能に相当する。

【００５２】

続いて、ＣＰＵ２２は、ＲＡＭ２５－２のワークエリアに格納されているプログラムソースモジュール４１の構文分析を行ない、そこに記述されているメソッドの定義をひとつ抽出する（Ｓ１０２）。ここで、ＣＰＵ２２は管理簿４２を参照し（Ｓ１０３）、抽出されたメソッド定義のメソッド名が管理簿４２の記録内容に含まれているか否かを判定する（Ｓ１０４）。この判定処理の結果がＹｅｓならばＳ１０５に進み、ＮｏならばＳ１０７に進む。

【００５３】

抽出されたメソッド定義のメソッド名が管理簿４２の記録内容に含まれているのであれば、ＣＰＵ２２は、プログラムソースモジュール４１が格納されていた

プログラムファイルの管理情報を調べ、プログラムソースモジュール４１の更新日時を取得する（Ｓ１０５）。そして、このプログラムソースモジュール４１の更新日時と、抽出されたメソッド定義のメソッド名の記録に対応付けられて管理簿４２に記録されている更新日時とが一致するか否かを判定する（Ｓ１０６）。この判定処理の結果がＹｅｓならばＳ１１１に進み、ＮｏならばＳ１０７に進む。

【００５４】

以上のＳ１０２からＳ１０６にかけて示されている処理は図３における高速化判定部３２の機能に相当する。

【００５５】

前述したＳ１０４またはＳ１０６のどちらかの判定処理でその結果がＮｏであったときには、ＣＰＵ２２は、Ｓ１０２に示した処理で抽出されたメソッド定義のコンパイルを実行し（Ｓ１０７）、その結果得られた、ＣＰＵ２２で直接実行可能なネイティブコードをＲＡＭ２５－２のワークエリアにコンパイル済み実行モジュール４４として格納し（Ｓ１０８）、更にそのネイティブコードをハードディスク装置２６にコンパイル済み格納モジュール４３として記憶させる（Ｓ１０９）。このネイティブコードの記憶は、プログラムソースモジュール４１の翻訳・実行が終了し、ＣＰＵ２２がこの図５に示したＪＩＴコンパイル処理の実行を一旦終了してもハードディスク装置２６に保持されている。

【００５６】

その後、ＣＰＵ２２は管理簿４２を更新し、前述したＳ１０２でコンパイルしたメソッド定義のメソッド名と、そのメソッド定義が記述されていたプログラムソースモジュール４１を格納していたプログラムファイルの管理情報に示されている更新日時とを対応付けて記録し（Ｓ１１０）、その後はＳ１１２に進む。

【００５７】

これらのＳ１０７からＳ１１０にかけて示されている処理は図３におけるコンパイル部３３の機能に相当する。

【００５８】

一方、前述したＳ１０６の判定処理でその結果がＹｅｓであったときには、Ｃ

ＰＵ２２は、Ｓ１０２に示した処理で抽出されたメソッド定義の翻訳であるネイティブコードをハードディスク装置２６から読み出し、読み出したそのネイティブコードをＲＡＭ２５－２のワークエリアにコンパイル済み実行モジュール４４として格納する（Ｓ１１１）。この処理は図３におけるファイルロード部３４の機能に相当する。

【００５９】

その後、ＣＰＵ２２はＲＡＭ２５－２のワークエリアにコンパイル済み実行モジュール４４として格納されているネイティブコードを直接実行する（Ｓ１１２）。この処理は図３におけるプログラム実行部３５の機能に相当する。

【００６０】

コンパイル済み実行モジュール４４を実行した後には、ＣＰＵ２２は、ＲＡＭ２５－２のワークエリアに格納されているプログラムソースモジュール４１の実行処理を終了したか否かを判定する（Ｓ１１３）。この判定処理の結果がＹｅｓならば今回のＪＩＴコンパイル処理を終了する。一方、この判定処理の結果がＮｏならばＳ１０２へ戻り、プログラムソースモジュール４１に次の実行順として記述されているメソッド定義について上述した処理を繰り返す。

【００６１】

以上までの処理が図５に示されているＪＩＴコンパイル処理である。

【００６２】

次に、本発明を実施するプログラム実行装置の第二の例について説明する。

【００６３】

この第二の例では、図２におけるハードディスク装置２６に、プログラムソースモジュール４１で記述されることがあり得るメソッド定義の翻訳であるネイティブコードを予め格納しておくようにするものである。こうすることにより、そのプログラムソースモジュール４１のプログラム実行装置１０上での初めての実行の際でも、プログラムソースモジュール４１に記述されているメソッド定義の翻訳であるネイティブコードをハードディスク装置２６から得ることのできる場合があるので、メソッドの翻訳処理に起因する実行開始時のタイムラグを短縮することができる。

【００６４】

本発明を実施するプログラム実行装置の第二の例の全体構成は図２に示したものと同様であるが、図２に示す構成要素のうち、ハードディスク装置２６の記憶内容が前述した第一の例と異なっている。図６は、本発明を実施するプログラム実行装置の第二の例におけるハードディスク装置の記憶内容を示す図であり、ハードディスク装置２６には、第一の例と同様のコンパイル済み格納モジュール４３が記憶されるのに加え、プログラムソースモジュール４１で記述されることがあり得るメソッド定義の翻訳であるコンパイル済み標準モジュール５１が予め記憶されている。このような、その翻訳であるネイティブコードがコンパイル済み標準モジュール５１としてハードディスク装置２６に記憶されるメソッド定義としては、例えば、Ｊａｖａの標準のクラスライブラリであるｊａｖａパッケージに含まれるクラスに属するメソッド定義などの、任意のＪａｖａ実行系で使用できることが保証されているメソッド定義が適用可能である。

【００６５】

次に図７について説明する。図７は本発明を実施するプログラム実行装置の第二の例における管理簿４２の例を表で示したものであり、ハードディスク装置２６の記憶内容が図６に示した内容である場合に対応するものである。図７を、前述した本発明の第一の実施例における管理簿４２の記録内容を示した図４と比較すると分かるように、図７においては、コンパイル済み格納モジュール４３についての管理情報が管理簿４２に記録される他に、コンパイル済み標準モジュール５１についての管理情報が管理簿に予め記録されている。コンパイル済み標準モジュール５１についての管理情報としては、そのコンパイル済み標準モジュール５１が翻訳である翻訳前のメソッド定義のメソッド名が管理簿４２に記録されている。

【００６６】

次に、本発明を実施するプログラム実行装置の第二の例におけるＣＰＵ２２により実行されるＪＩＴコンパイル処理の処理内容について、図８に示すフローチャートを参照しながら説明する。

【００６７】

まず、図８におけるＳ２０１からＳ２０４にかけて示されている処理は、図５にフローチャートで示したＪＩＴコンパイル処理の第一の例におけるＳ１０１からＳ１０４にかけて示した処理と全く同一のものであり、その説明は省略する。

【００６８】

Ｓ２０５において、ＣＰＵ２２は、Ｓ２０２で抽出されたメソッド定義のメソッド名が管理簿４２のコンパイル済み標準モジュール５１についての管理情報として記録されているか否かを判定する。この判定処理の結果がＹｅｓならばＳ２１２に進み、ＮｏならばＳ２０６に進む。

【００６９】

図８におけるＳ２０５以降の、Ｓ２０６からＳ２１４にかけて示されている処理も、図５にフローチャートで示したＪＩＴコンパイル処理の第一の例におけるＳ１０５からＳ１１３にかけて示した処理と全く同一のものである。

【００７０】

次に、本発明を実施するプログラム実行装置の第三の例について説明する。

【００７１】

この第三の例では、プログラム実行装置に電源を投入して起動させたときに、ハードディスク装置２６の記憶内容をＲＡＭ２５－２に複写して格納する。そして、ＣＰＵ２２によって行なわれるＪＩＴコンパイル処理では、ハードディスク装置２６の記憶内容に基づいて翻訳・実行の処理を進めるのではなく、ハードディスク装置２６の記憶内容の複写であるＲＡＭ２５－２の格納データに従ってその処理を進める。一般的に、記憶内容の読み出しは半導体メモリからの方がハードディスク装置よりも高速に行なえるので、こうすることにより、プログラム装置１０によるプログラムソースモジュール４１の翻訳・実行の処理時間を更に短縮することができる。

【００７２】

本発明を実施するプログラム実行装置の第三の例の全体構成は図２に第一の例として示したものと同様であるが、図９に示すように、ＲＡＭ２５－２の記憶領域に、ＣＰＵ２２による処理の実行のために一時的に使用されるワークエリア６１の領域の他に、ハードディスク装置２６の記憶内容が複写されて格納されるキ

ャッシュエリア６２の領域を設けるようにする。そして、このキャッシュエリア６２の領域は、ＣＰＵ２２が後述するＪＩＴコンパイル処理の実行を終了してもその記憶内容をクリアせずに保持するようにし、次にＪＩＴコンパイル処理を改めて実行しても、その処理においてその記憶内容が利用できるようにする。

【００７３】

また、管理簿４２の記録内容も図４に示したものと同様のものでよい。但し、前述した第一の例では管理簿４２はハードディスク装置２６に記憶されていたが、この第三の例では、ハードディスク装置２６の管理簿４２についての記録内容もＲＡＭ２５－２に複写され、ＣＰＵ２２はＲＡＭ２５－２上の管理簿４２を参照して処理を行なうものとする。

【００７４】

本発明を実施するプログラム実行装置の第三の例における、装置全体の制御処理の処理内容を図１０にフローチャートで示す。この処理は、本装置２０に電源を投入するとその直後にＣＰＵ２２により直ちに実行される処理である。

【００７５】

本装置に電源が投入されると、まず、ＣＰＵ２２は初期化処理を実行する（Ｓ３０１）。初期化処理はＣＰＵ２２自身の有する各種のレジスタを初期化したり、ＲＡＭ２５－２の初期化などを行なう処理である。

【００７６】

続いて、ＣＰＵ２２はハードディスク装置２６の記憶内容をＲＡＭ２５－２のキャッシュエリア６２に複写する（Ｓ３０２）。

【００７７】

その後、ＣＰＵ２２は、本装置２０の入力部２１、出力部２３、Ｉ／Ｆ部２４の制御等を含む各種の制御処理を実行し（Ｓ３０３）、その後にＪＩＴコンパイル処理を実行する（Ｓ３０４）。この処理を終えた後にはＳ３０３へ戻り、以降の処理が繰り返される。

【００７８】

次に図１１について説明する。図１１は、図１０にＳ３０４として示されているＪＩＴコンパイル処理の処理内容を示すフローチャートである。

【００７９】

まず、図８におけるＳ３１１からＳ３１９にかけて示されている処理は、図５にフローチャートで示したＪＩＴコンパイル処理の第一の例におけるＳ１０１からＳ１０９にかけて示した処理と同様のものである。但し、Ｓ３１３において、ＣＰＵ２２は、ＲＡＭ２５－２のキャッシュエリア６２に格納されている、ハードディスク装置２６の複写である管理簿４２を参照する。

【００８０】

Ｓ３２０において、Ｓ３１７でのコンパイルにより得られたネイティブコードをＲＡＭ２５－２のキャッシュエリア６２にも格納し、このネイティブコードをＳ３１９において記憶させたハードディスク装置２６の記憶内容とキャッシュエリア６２の記憶内容とを一致させる。

【００８１】

その後、ＣＰＵ２２は、ハードディスク装置２６及びキャッシュエリア６２双方の管理簿４２を同様に更新し、Ｓ３１２でコンパイルしたメソッド定義のメソッド名と、そのメソッド定義が記述されていたプログラムソースモジュール４１を格納していたプログラムファイルの管理情報に示されている更新日時とを対応付けて記録し（Ｓ３２１）、その後はＳ３２３に進む。

【００８２】

一方、Ｓ３１６の判定処理でその結果がＹｅｓであったときには、ＣＰＵ２２は、Ｓ３１２に示した処理で抽出されたメソッド定義の翻訳であるネイティブコードをＲＡＭ２５－２のキャッシュエリア６２から読み出し、読み出したそのネイティブコードをＲＡＭ２５－２のワークエリアにコンパイル済み実行モジュール４４として格納する（Ｓ３２２）。

【００８３】

図１１におけるＳ３２２以降の、Ｓ３２３及びＳ３２４に示されている処理は、図５にフローチャートで示したＪＩＴコンパイル処理の第一の例におけるＳ１１２及びＳ１１３に示した処理と全く同一のものであり、説明は省略する。

【００８４】

なお、前述した第二の例、すなわちハードディスク装置２６に、プログラムソ

ースモジュール４１で記述されることがあり得るメソッド定義の翻訳であるネイティブコードを予め格納しておくようにしたプログラム実行装置で、この第三の例のように、ハードディスク装置２６の記憶内容をＲＡＭ２５－２に複写してＪＩＴコンパイル処理を実行することも可能である。

【００８５】

なお、上述した本発明の各実施形態において説明したＪＩＴコンパイル処理を汎用的なコンピュータで実施させることも可能である。そのためには、本発明の各実施形態において説明した、図５、図８、あるいは図１１のＪＩＴコンパイル処理に相当する処理をコンピュータに行なわせる制御プログラムをそのコンピュータで読み取り可能な記録媒体（記憶媒体）に予め記憶させておき、その記録媒体から読み出したその制御プログラムを読み出させてそのコンピュータのメインメモリに一旦格納させた後に、そのコンピュータの有する中央処理装置に格納されたこのプログラムを読み出させて実行させるように構成すればよい。

【００８６】

上述した制御プログラムを格納し、且つそれをコンピュータで読み取ることの可能な記録媒体の例を図１２に示す。このような記録媒体としては、例えば、コンピュータ７１の本体に内蔵若しくは外付けされる半導体メモリやハードディスク装置などのメモリ７２、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、ＭＯ（光磁気ディスク）、フロッピーディスクなどといった可搬型記憶媒体７３、あるいはコンピュータ７１と回線７４で接続されていてコンピュータ７１がプログラムをダウンロードすることの可能なプログラムサーバ７５の記憶装置７６などがあるが、これらのいずれであってもよい。

【００８７】

【発明の効果】

以上詳細に説明したように、本発明は、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させるときに、前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、電源電圧が消失しても記憶内容が保持される記憶部に記憶させ、前記原始プログラ

ムに記述されている関数の翻訳である前記機械語が前記記憶部に記憶されているか否かの判定を行ない、前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または前記記憶部に記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させるように構成する。そして、この構成により、電源を切った後に同一の原始プログラムを再度実行させるような場合には、プログラムの翻訳処理に起因する実行開始時のタイムラグを生じさせずに原始プログラムを実行させることができる。

【００８８】

あるいは、本発明は、ジャスト・イン・タイム・コンパイラ方式により、原始プログラムを特定の演算処理システムのプラットホームで直接実行可能な機械語に翻訳して該機械語を実行させるときに、前記原始プログラムの翻訳である前記機械語を、該原始プログラムに記述されている関数ごとに、該機械語の翻訳前の原始プログラムが更新された日時に対応付けて記憶し、前記原始プログラムの更新された日時と、記憶されている前記機械語に対応付けられている更新日時とが一致するか否かの判定を行ない、前記判定結果に応じて、前記原始プログラムを翻訳して得る機械語、または記憶されている機械語、のどちらかを特定の演算処理システムのプラットホームで直接実行させるように構成する。そして、この構成により、こうすることによって、原始プログラムに変更が加えられても、プログラムの翻訳処理に起因する実行開始時のタイムラグを生じさせずに、且つ、その変更に応じて正しく原始プログラムを実行させることができる。

【００８９】

以上のように、本発明のいずれの構成によっても、ＪＩＴコンパイラを用いて原始プログラムを実行させるときの実行開始時の性能を向上させることができるという効果を奏する。

【図面の簡単な説明】

【図１】

本発明の基本構成図である。

【図２】

本発明を実施するプログラム実行装置の全体構成を示す図である。

【図３】

本発明を実施するプログラム実行装置における翻訳・実行処理の全体の流れを示した図である。

【図４】

管理簿の一例を表で示した図である。

【図５】

ＪＩＴコンパイル処理の第一の例の処理内容をフローチャートで示した図である。

【図６】

本発明を実施するプログラム実行装置の第二の例におけるハードディスク装置の記憶内容を示す図である。

【図７】

プログラム実行装置の第二の例における管理簿の例を表で示した図である。

【図８】

ＪＩＴコンパイル処理の第二の例の処理内容をフローチャートで示した図である。

【図９】

本発明を実施するプログラム実行装置の第三の例におけるＲＡＭの記憶内容を示す図である。

【図１０】

本発明を実施するプログラム実行装置の第三の例における装置全体の制御処理の処理内容を示すフローチャートである。

【図１１】

図１０におけるＪＩＴコンパイル処理の処理内容を示すフローチャートである。

【図１２】

記憶させた制御プログラムをコンピュータで読み取ることの可能な記録媒体の例を示す図である。

【符号の説明】

１　原始プログラム

１０、２０　プログラム実行装置

１１　記憶手段

１２　翻訳手段

１３　記憶制御手段

１４　判定手段

１５　実行制御手段

１６　実行手段

２１　入力部

２２　ＣＰＵ

２３　出力部

２４　Ｉ／Ｆ部

２５　半導体メモリ

２５－１　ＲＯＭ

２５－２　ＲＡＭ

２６　ハードディスク装置

２７　バス

３０　ＪＩＴコンパイラ

３１　プログラムロード部

３２　高速化判定部

３３　コンパイル部

３４　ファイルロード部

３５　プログラム実行部

４１　プログラムソースモジュール

４２　管理簿

４３　コンパイル済み格納モジュール

４４　コンパイル済み実行モジュール

５１　コンパイル済み標準モジュール

６１　ワークエリア

６２　キャッシュエリア

７１　コンピュータ

７２　メモリ

７３　可搬型記憶媒体

７４　回線

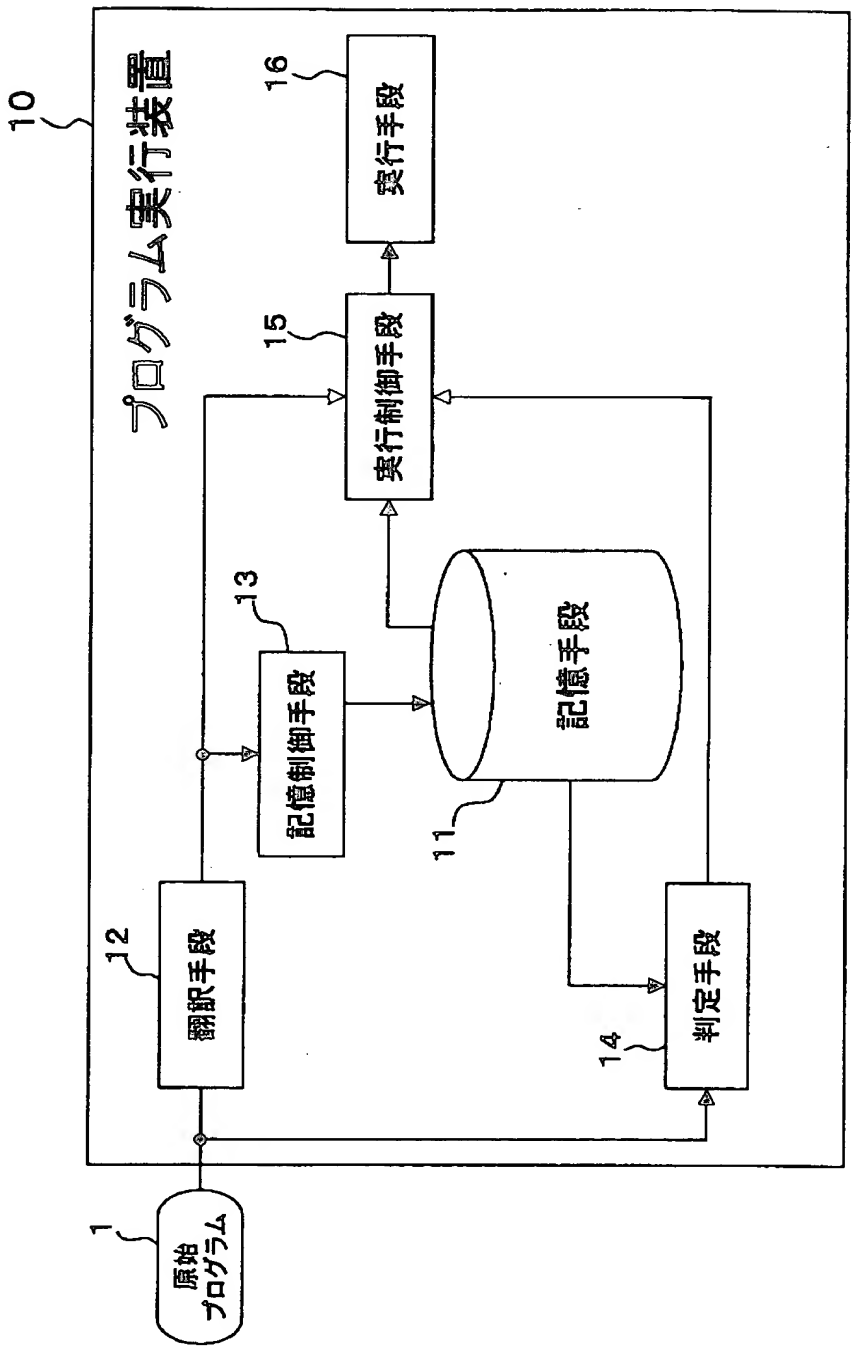
７５　プログラムサーバ

７６　記憶装置

【書類名】　　図面

【図１】

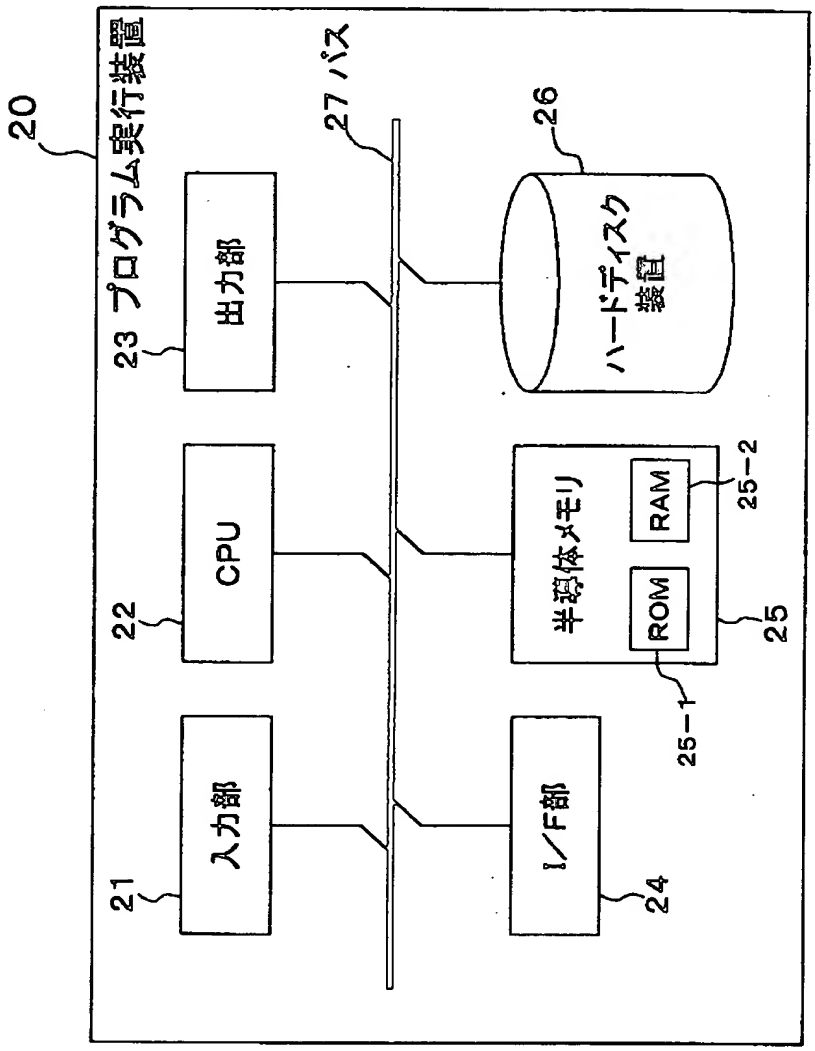
本発明の基本構成図



出証特２０００－３０７１４２４

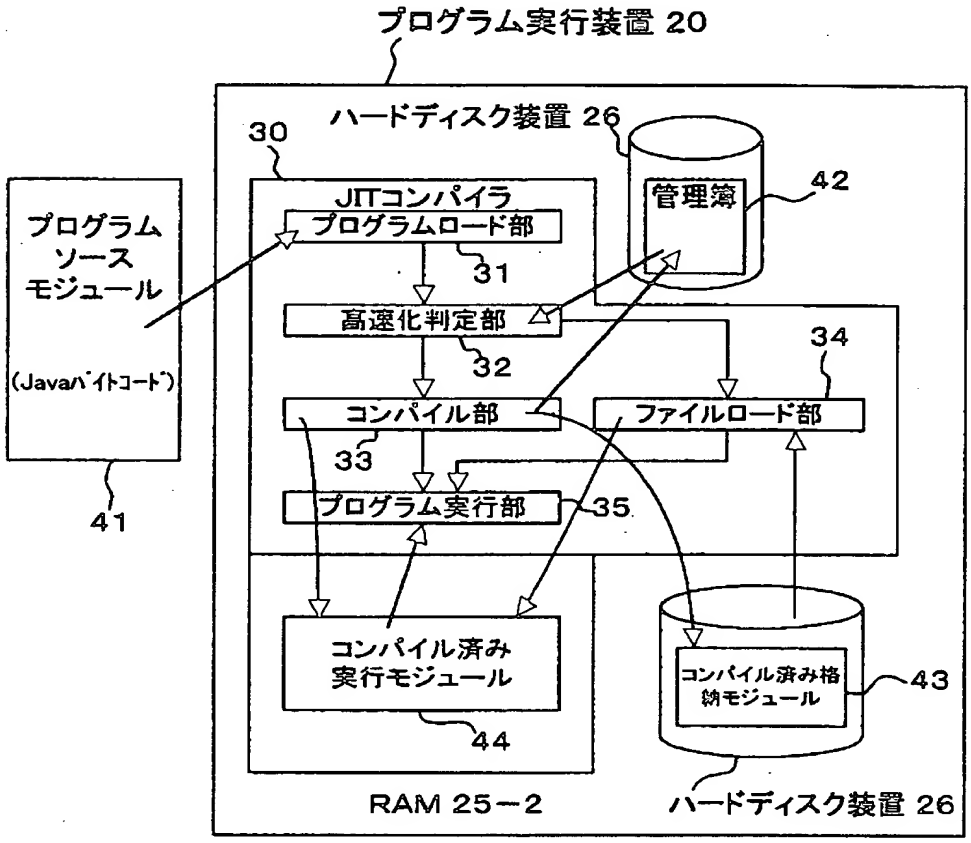
【図２】

本発明を実施するプログラム実行装置の全体構成を示す図



【図３】

本発明を実施するプログラム実行装置における翻訳・実行処理の全体の流れを示した図



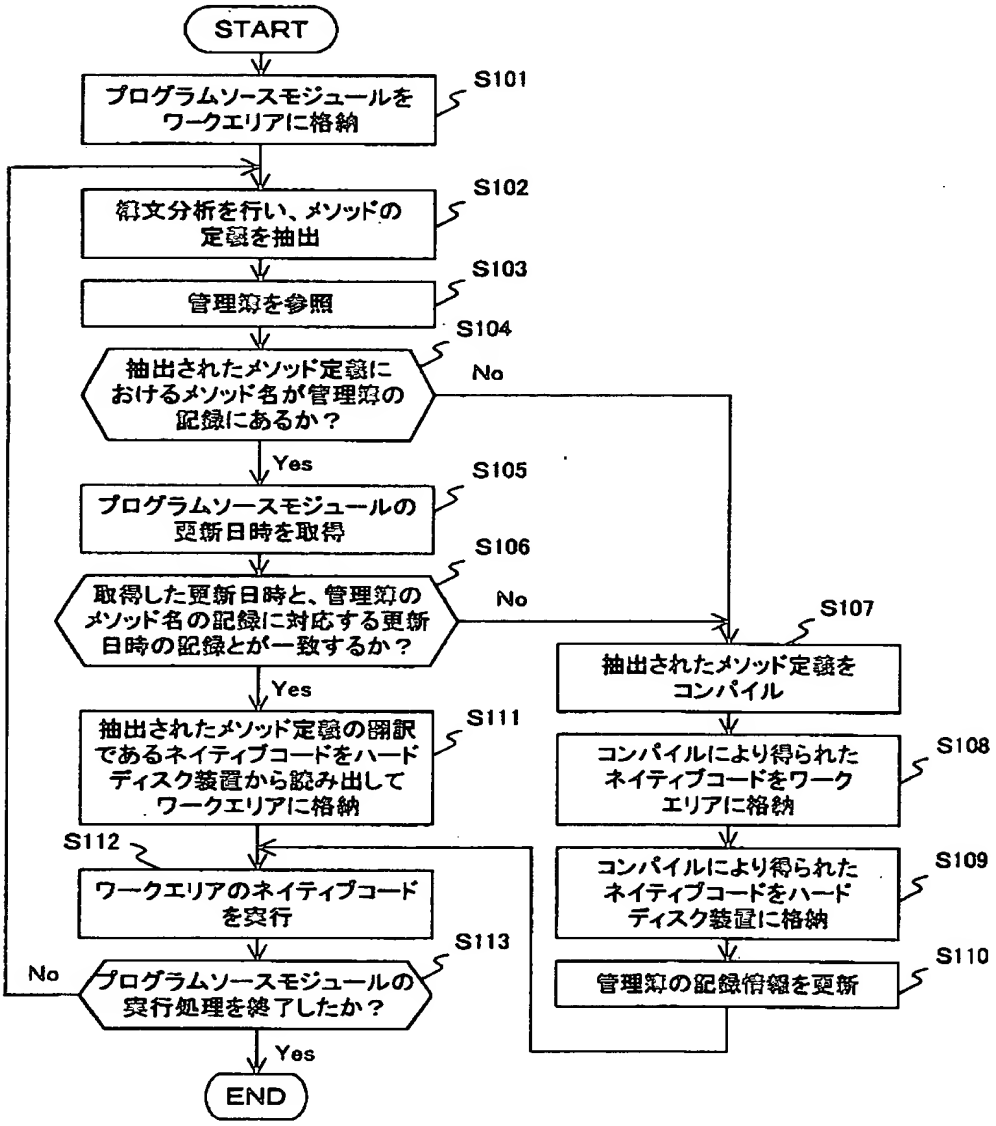
【図４】

管理簿の一例を表で示した図

メソッド名（コンパイル対象名）	プログラムソースモジュール更新日時
open	1999. 09. 10 : 13. 25. 00
read	2000. 01. 11 : 09. 57. 34
write	2000. 01. 11 : 09. 57. 34
close	1999. 09. 10 : 13. 25. 00

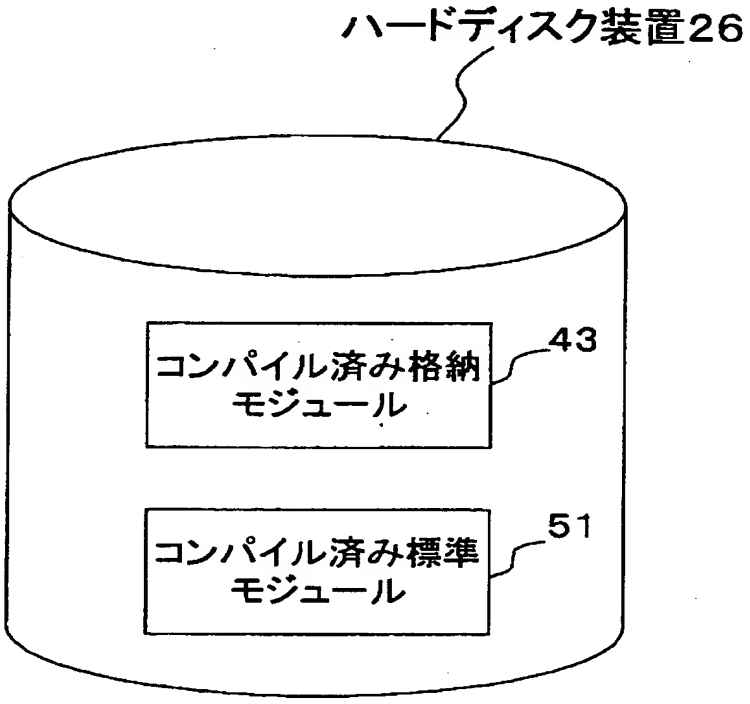
【図５】

JITコンパイル処理の第一の例の処理内容をフローチャートで示した図



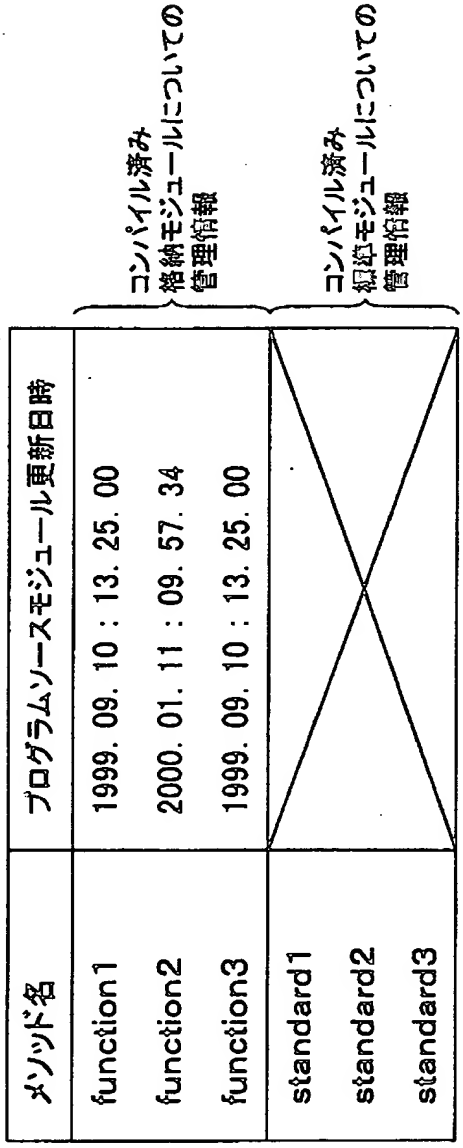
【図６】

本発明を実施するプログラム実行装置の第二の例におけるハードディスク装置の記憶内容を示す図



【図７】

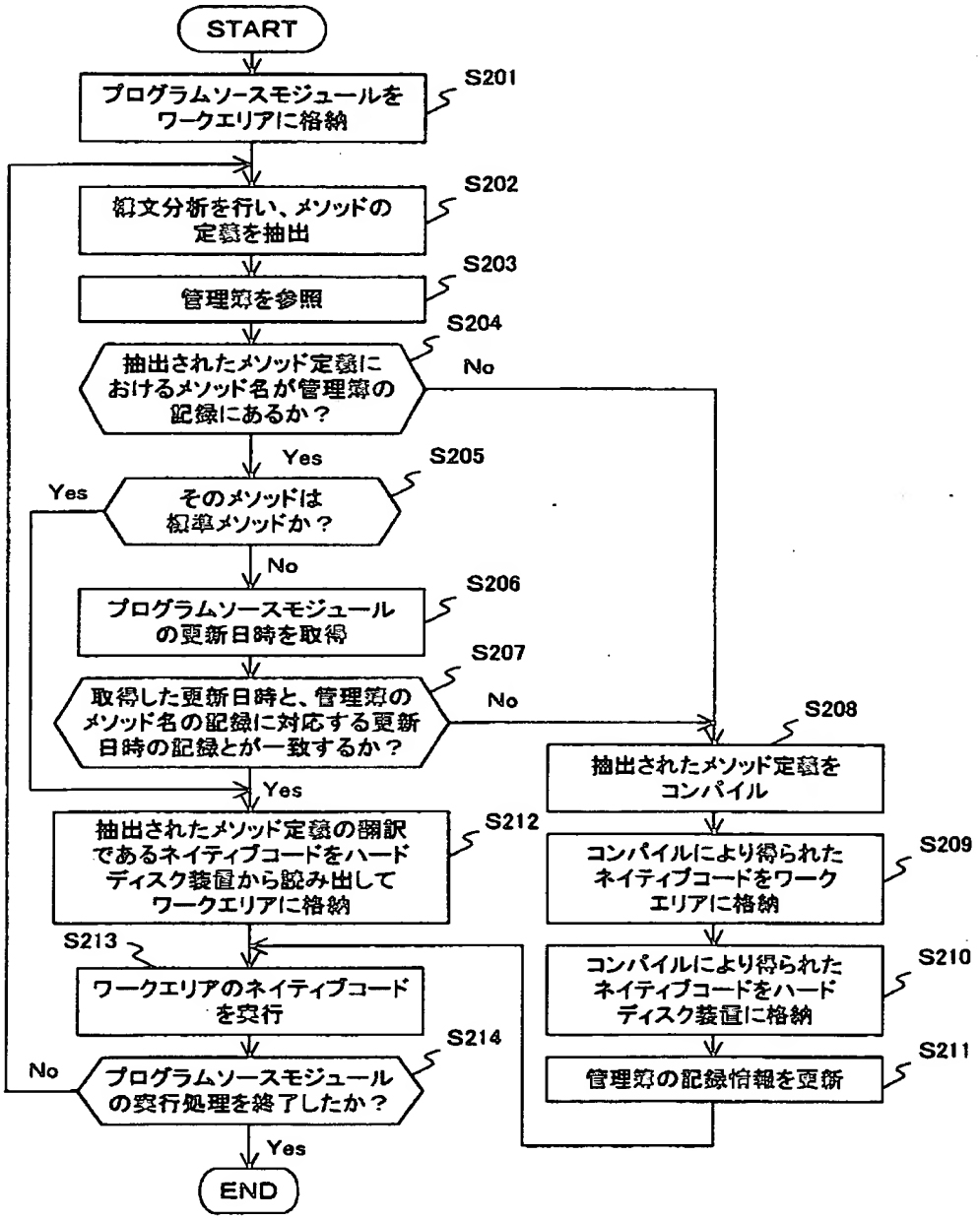
プログラム実行装置の第二の例における管理簿の例を表で示した図



メソッド名	プログラムソースモジュール更新日時
function1	1999. 09. 10 : 13. 25. 00
function2	2000. 01. 11 : 09. 57. 34
function3	1999. 09. 10 : 13. 25. 00
standard1	
standard2	
standard3	

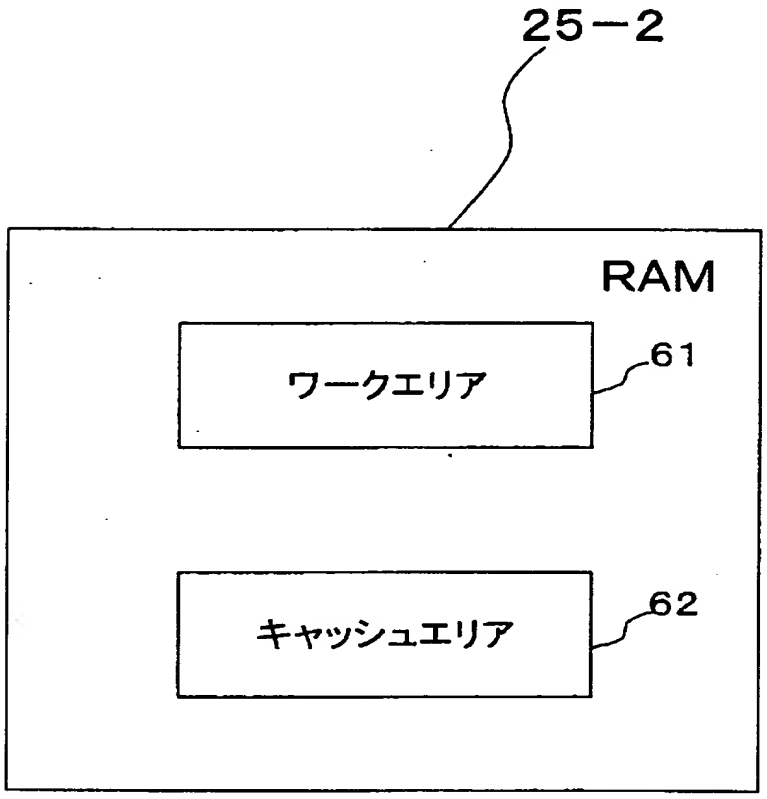
【図８】

JITコンパイル処理の第二の例の処理内容をフローチャートで示した図



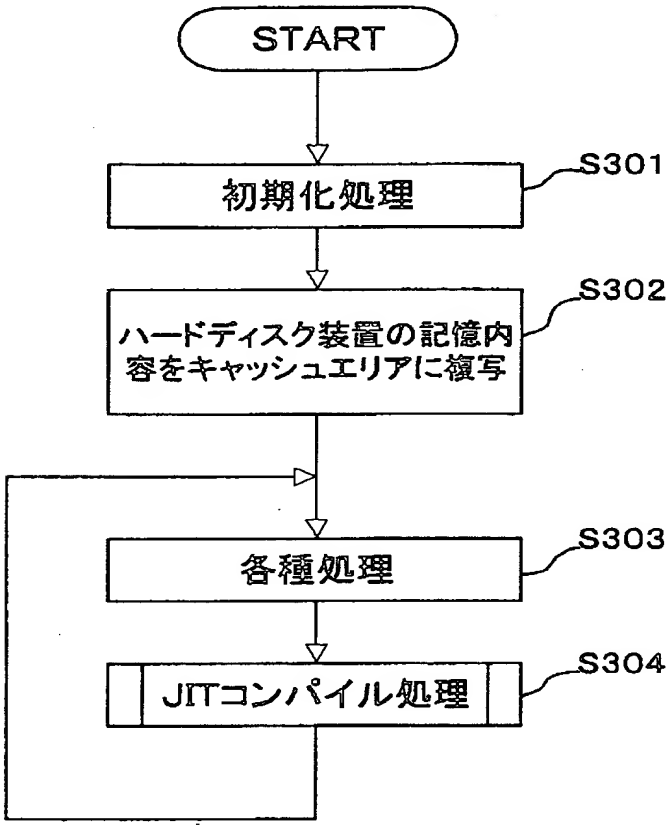
【図９】

本発明を実施するプログラム実行装置の
第三の例におけるRAMの記憶内容を示す図



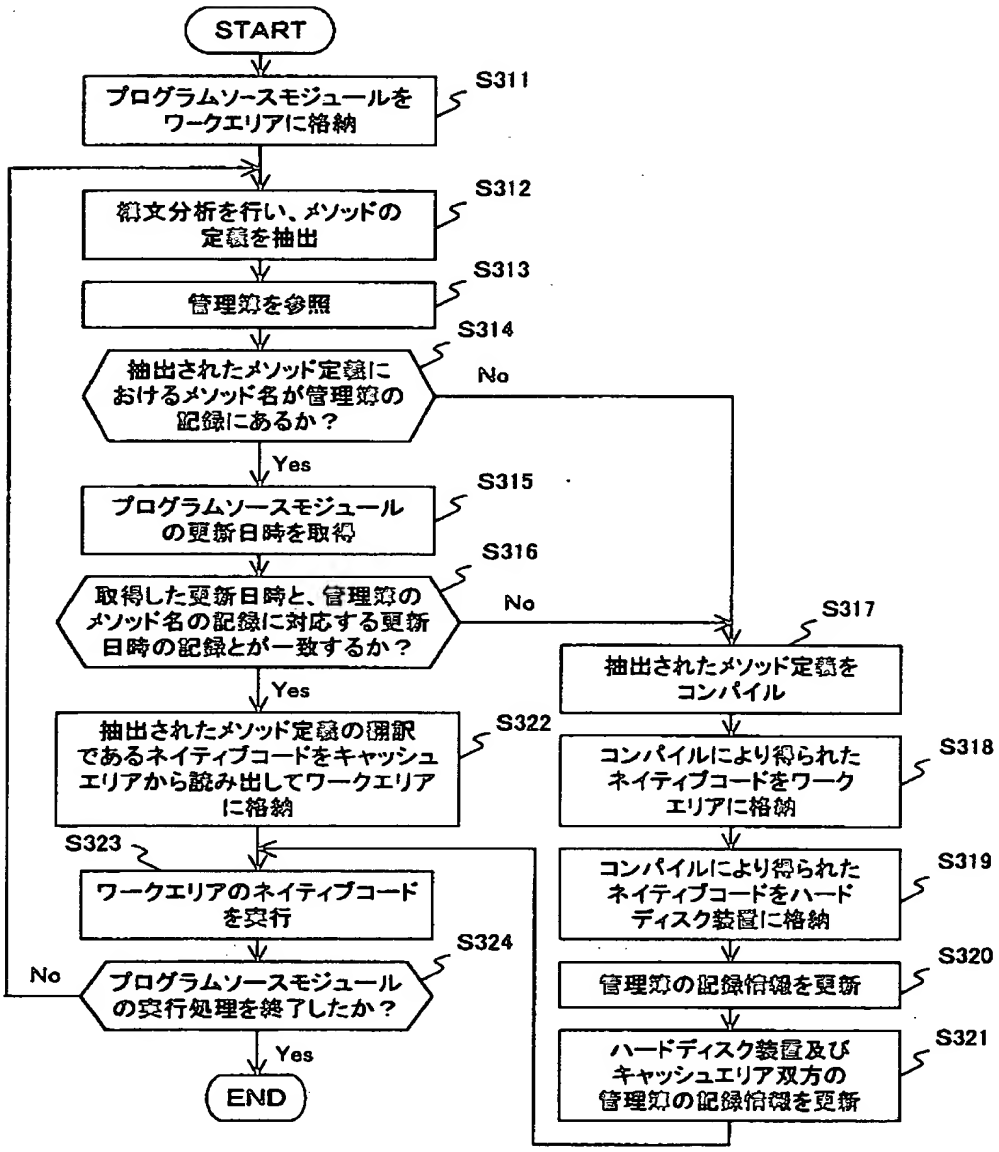
【図１０】

本発明を実施するプログラム実行装置の第三の例における
装置全体の制御処理の処理内容を示すフローチャート



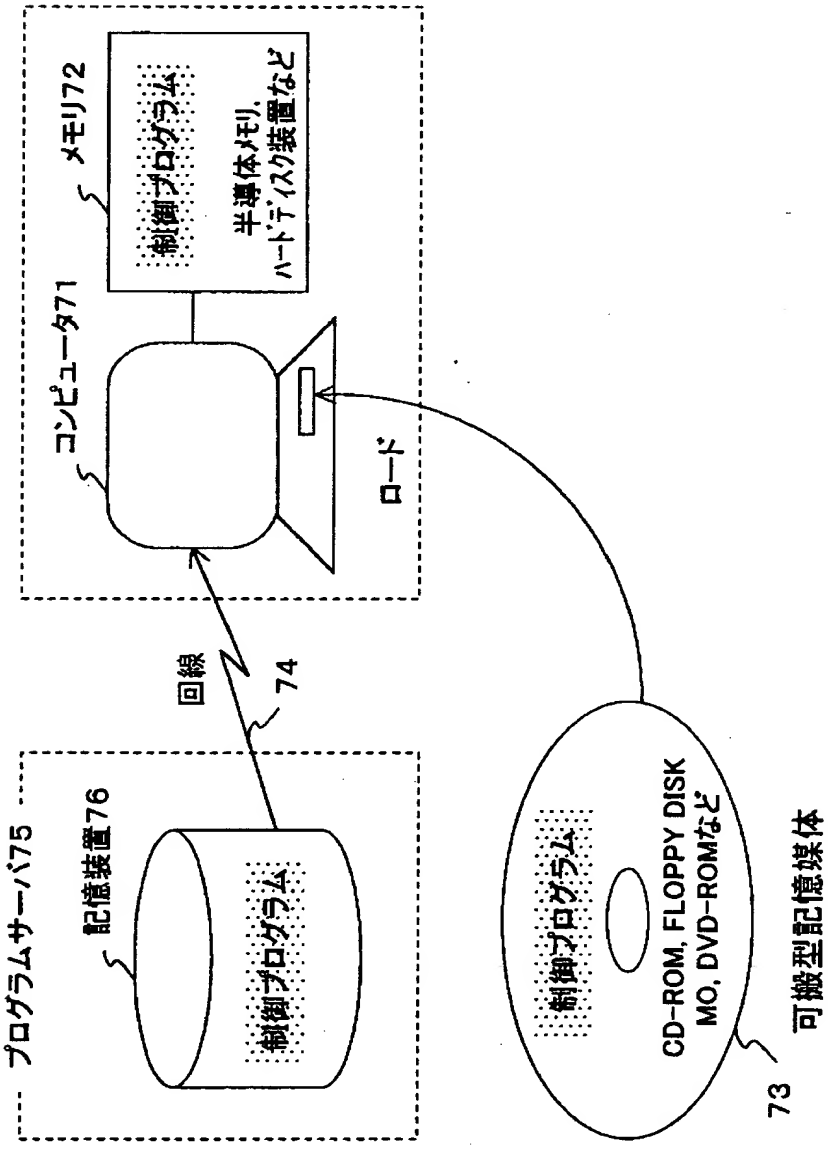
【図１１】

図１０におけるＪＩＴコンパイル処理の処理内容をフローチャート



【図１２】

記憶させた制御プログラムをコンピュータで
読み取ることの可能な記録媒体の例を示す図



【書類名】　　要約書

【要約】

【課題】　ＪＩＴコンパイラを用いて原始プログラムを実行させるときの実行開始時の性能を向上させる。

【解決手段】　プログラム実行装置１０に、原始プログラム１に記述される関数の翻訳であって実行手段１６で実行可能な機械語を該関数ごとに記憶する、電源電圧が消失しても記憶内容が保持される記憶手段１１と、原始プログラム１を実行手段１６で実行可能な機械語に翻訳する翻訳手段１２と、翻訳手段１２により翻訳された機械語を記憶手段１１に記憶させる記憶制御手段１３と、原始プログラム１で用いられている関数の翻訳である機械語が記憶手段１１に記憶されているか否かの判定を行なう判定手段１４と、判定手段１４による判定結果に応じて、翻訳手段１２の翻訳する機械語、または記憶手段１１に記憶されている機械語、のどちらかを実行手段１６に直接実行させる実行制御手段１５とを備える。

【選択図】　　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社